DETAILED ACTION

1. Claims 1, 2, 4-12, 14-20, 22-24, 26-32, 34, 35 and 37-42.

Examiner's Amendment

2. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Gerhard W. Thielman on 12/5/2008. The examiner's amendment is necessitated to further clarify the claimed invention and to distinguish from the prior arts of record.

Claims 1, 17, and 34 have been further amended below:

Claim 33 has been cancelled.

1. (Currently Amended) A computer-implemented method comprising:

providing asynchronous access to multiple users to a graphical programming and analysis environment program visually represented as a white board;

wherein each user of the multiple users generates graphically represented code objects within the environment program, to further comprise:

said each user instantiates one or more code objects,

said each user determines an internal logic for each code object of said one or more code objects,

said each user determines first data to be received by said each code object, and

said each user determines second data to be sent by said each code object;

wherein said each user obtains access to the code objects of other users of the multiple users based on security privileges accorded to said each user, in which the code objects reported over a network are hierarchically filtered based on the security privileges according to said each user, and the security privileges restrict those functions or features of the code objects available to said each user;

wherein said each user chains the code objects of said each user to the code objects of the other users to which said each user has access to yield inter-code object communication by inter-code object connections, each inter-code object connection terminating on one of an edge and an interior of one of the code objects and,

wherein said each user executes an application program composed of the code objects as chained together within the environment program, the program operating by:

opening a first input window that displays a first dialog box and a first acknowledgement cursor region, wherein the first dialog box receives the first data and the internal logic receives the first data in response to said each user executing the first acknowledgement cursor region,

opening a second input window that displays a second dialog box and a second acknowledgement cursor region, wherein the second dialog box receives the second data, and the internal logic receives the second data in response to said each user executing the second acknowledgement cursor region, and

opening an output window that displays result data from the internal logic operating on the first and second data, wherein the result data reported over the network are hierarchically filtered based on the security privileges;

opening a chat area within which said each user can communicate with other users; and

opening a user list area that displays a name of the multiple users that are currently logged into the environment program.

17. (Currently Amended) An apparatus to provide an environment for multiple-user graphical programming and analysis by machine-readable instructions executable on a computer platform said: apparatus comprising:

at least one or more processors configured to:

create a plurality of graphically represented code objects, each code object created by a user and accessible by other users in accordance with security privileges of the other users, said each code object comprises:

a data interface indicating first data to be input into the code object and second data to be output by the code object, and

internal logic to generate the second data from the first data;

create a plurality of graphically represented inter-code object connections, each inter- code object connection representing data transfer between a pair of code objects;

create at least one application program composed of one or more chains of the code objects interconnected via the inter-code object connections, the program including opening operations for:

a first input window for playing a first dialog box and a first acknowledgment cursor region, wherein the first dialog box receives the first data and the internal logic receives the first data in response to said each user executing the first acknowledgement cursor region,

a second input window for displaying a second dialog box and a second acknowledgement cursor region, wherein the second dialog box receives the second data, and the internal logic receives the second data in response to said each user executing the second acknowledgement cursor region, and

an output window for displaying result data from the internal logic operating on the first and second data, wherein the result data reported over the network are hierarchically filtered based on the security privileges; and,

create a graphical white board area within which to dispose the code objects and to create the inter-code object connections,

create a chat area within which the said user can communicate with other users; and

create a user list area showing a name of each of the user and other users currently logged into the environment program,

wherein the application program is executable within the graphical white board area, and each inter-code object connection terminates on one of an edge and an interior of one of the code object, and

wherein said each user obtains access to the code objects of other users based on security privileges accorded to the user, in which the code objects reported over a network area hierarchically filtered based on the security privileges according to the user, and the security privileges restrict those functions or features of the code objects available to the user.

34. (Currently Amended) A computer-implemented method comprising:

accessing by a user a graphical programming and analysis environment program that other users are already currently accessing;

generating by the user graphically represented code objects within the environment program, wherein for each code object,

the user determining a data interface indicating first data to be input into the code object and second data to be output by the code object; and

the user determining internal logic to generate the second data from the first data;

graphically chaining together the code objects by the user within the environment program, including chaining together the code objects generated by the user and code objects generated by the other users to which the user has access based on security privileges accorded to the user, to yield inter-code object communication by inter-code object connections, each inter- code object connection terminating on one of an edge and an interior of one of the code objects;

hierarchically filtering the code objects reported over a network based on the security privileges according to the user; and

assembling an application program by the user within the environment program, each application prepare composed of the code objects as have been chained together the application program operating by:

opening a first input window that displays a first dial0gbox and a first acknowledgement cursor region, wherein the first dialog box receives the first data, and the internal logic receives the first data in response to said each user executing the first acknowledgement cursor region,

opening a second input window that displays a second dialog box and a second acknowledgement cursor region, wherein the second dialog box receives the second data and the internal logic receives the second data in response to said each user executing the second acknowledgement cursor region, and

opening an output window that displays result data from the internal logic operating on the first and second data, wherein the result data reported over the network are hierarchically filtered based on the security privileges;

opening a chat area within which said each user can communicate with other users;

opening a user list area that displays a name of multiple users that are currently logged into the environment program, and

providing access for each said user to the code objects of the other users of the multiple users based on security privileges accorded to the user, wherein the security privileges restrict those functions or features of the code objects available to the user.

Examiner's Statement of Reasons for Allowance

3. The following is an examiner's statement of reasons for allowance:

The prior art of record (i.e. Dye et al.) taken alone or in combination with other prior arts fails to teach or reasonably suggest *wherein said each user obtains access to the code objects of other users of the multiple users based on security privileges accorded to said each user, in which the code objects reported over a network are hierarchically filtered based on the security privileges according to said each user, and the security privileges restrict those functions or features of the code objects available to said each user; and where the results data reported over the network area hierarchically filtered based on the security privileges* as recite in the independent claims.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Phillip H. Nguyen whose telephone number is (571) 270-1070. The examiner can normally be reached on Monday - Thursday 10:00 AM - 3:00 PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

PN
12/6/2008
/Wei Y Zhen/
Supervisory Patent Examiner, Art Unit 2191